

# CADP 2020

## PRÁCTICA 6 – ALOCACIÓN DINÁMICA - PUNTEROS

### Parte conceptual

- 1) ¿Qué es la memoria estática?
- 2) ¿Qué es la memoria dinámica?
- 3) ¿Qué es una variable del tipo puntero?
- 4) ¿Qué hace la operación de NEW sobre una variable del tipo puntero?
- 5) ¿Qué hace la operación de DISPOSE sobre una variable del tipo puntero?

### Parte Práctica

Para los ejercicios de la parte práctica, utilizaremos la función de Pascal **sizeof**, que recibe como parámetro una variable de cualquier tipo y retorna el espacio que dicha variable ocupa en la memoria principal.

Para realizar estos ejercicios, considere la siguiente tabla, que representa la cantidad de bytes que ocupa la representación interna de distintos tipos de datos en un compilador Pascal típico.

Tipo	Cantidad de bytes
Entero	2 bytes
Real	4 bytes
Char	1 byte
String	Tantos bytes como indique la longitud del String + 1
Record	La suma de las longitudes de los campos del registro
Puntero	4 bytes
Boolean	1 byte

- 1) Indique los valores que imprime el siguiente programa en pascal.

```
program punteros;
type
  cadena = string[50];
  puntero_cadena = ^cadena;
var
  pc: puntero_cadena;
begin
  writeln(sizeof(pc), ' bytes');
  new(pc);
  writeln(sizeof(pc), ' bytes');
  pc^:= 'un nuevo nombre';
  writeln(sizeof(pc), ' bytes');
  writeln(sizeof(pc^), ' bytes');
  pc^:= 'otro nuevo nombre distinto al anterior';
  writeln(sizeof(pc^), ' bytes');
```

**end.**

- 2) Indique los valores que imprime el siguiente programa en pascal.

```
program punteros;
type
  cadena = string[9];
  producto = record
    codigo: integer;
    descripcion: cadena;
    precio: real;
  end;
  puntero_producto = ^producto;
var
  p: puntero_producto;
  prod: producto;
begin
  writeln(sizeof(p), ' bytes');
  writeln(sizeof(prod), ' bytes');
  new(p);
  writeln(sizeof(p), ' bytes');
  p^.codigo := 1;
  p^.descripcion := 'nuevo producto';
  writeln(sizeof(p^), ' bytes');
  p^.precio := 200;
  writeln(sizeof(p^), ' bytes');
  prod.codigo := 2;
  prod.descripcion := 'otro nuevo producto';
  writeln(sizeof(prod), ' bytes');
end.
```

- 3) Indique los valores que imprime el siguiente programa en pascal.

```
program punteros;
type
  numeros = array[1..10000] of integer;
  puntero_numeros = ^numeros;
var
  n: puntero_numeros;
  num: numeros;
  i: integer;
begin
  writeln(sizeof(n), ' bytes');
  writeln(sizeof(num), ' bytes');
  new(n);
  writeln(sizeof(n^), ' bytes');
  for i:= 1 to 5000 do
    n^[i] := i;
  writeln(sizeof(n^), ' bytes');
end.
```

4) Indique los valores que imprimen los siguientes programas en Pascal.

a) **program** punteros;  
  **type**  
    cadena = string[50];  
    puntero\_cadena = ^cadena;  
  **var**  
    pc: puntero\_cadena;  
  **begin**  
    pc^:= 'un nuevo texto';  
    new(pc);  
    writeln(pc^);  
  **end.**

b) **program** punteros;  
  **type**  
    cadena = string[50];  
    puntero\_cadena = ^cadena;  
  **var**  
    pc: puntero\_cadena;  
  **begin**  
    new(pc);  
    pc^:= 'un nuevo nombre';  
    writeln(sizeof(pc^), ' bytes');  
    writeln(pc^);  
    dispose(pc);  
    pc^:= 'otro nuevo nombre';  
  **end.**

c) **program** punteros;  
  **type**  
    cadena = string[50];  
    puntero\_cadena = ^cadena;  
  **procedure** cambiarTexto(pun: puntero\_cadena);  
  **begin**  
    pun^:= 'Otro texto';  
  **end;**  
  **var**  
    pc: puntero\_cadena;  
  **begin**  
    new(pc);  
    pc^:= 'Un texto';  
    writeln(pc^);  
    cambiarTexto(pc);  
    writeln(pc^);  
  **end.**

d) **program** punteros;  
  **type**  
    cadena = string[50];

```
puntero_cadena = ^cadena;  
procedure cambiarTexto(pun: puntero_cadena);  
begin  
    new(pun);  
    pun^:= 'Otro texto';  
end;  
var  
    pc: puntero_cadena;  
begin  
    new(pc);  
    pc^:= 'Un texto';  
    writeln(pc^);  
    cambiarTexto(pc);  
    writeln(pc^);  
end.
```